A Mini-Project Report on

STOCK MARKET MANAGEMENT SYSTEM

Submitted by

GOURAV KUMAR SHARMA (U03NM21T029013)

SUDHANSHU SHARMA (U03NM21TO29061)

V SEM, B. TECH (CSE)

Under the guidance of

Dr. Champa H N
Professor,
Department of CSE,
UVCE

Punith Kumar
Research Scholar,
Department of CSE,
UVCE

Department of Computer Science and Engineering

UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

K.R. Circle, Bangalore – 560001

UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

K.R. Circle, Bangalore – 560001

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **GOURAV KUMAR SHARMA** of V Semester, B.Tech, Computer Science and Engineering, bearing the register number **U03NM21T029013** has submitted the DBMS Mini-Project Report on **"STOCK MARKET MANAGEMENT SYSTEM"**, in partial fulfilment for the DBMS Lab, prescribed by the Bangalore University for the academic year 2022-23.

Mr. Punith Kumar	**Dr. Champa H N**	**Dr. Thriveni J**
Reasearch Scholar,	**Professor,**	**Chairperson,**
Dept. of CSE,	Dept. of CSE	Dept. of CSE,
UVCE	UVCE	UVCE

Examiners:

1. …… 2. ...…….

ACKNOWLEDGEMENT

I take this opportunity to thank our institution **University Visvesvaraya College of Engineering** for having given me an opportunity to carry out this project.

I would like to thank **Dr. Paul S Vizhian, Principal, UVCE**, for providing us all the facilities to work on this project. I am indebted to him for being my pillar of strength and inspiration.

I wish to place my grateful thanks to **Dr. Thriveni J, Professor and Chairperson, Department of Computer Science and Engineering, UVCE,** who helped me to make my project a great success.

I am grateful to **Dr. Champa H N, Professor, Department of Computer Science and Engineering**, **UVCE,** for her valuable suggestions and support, which has sustained me throughout the course of the project.

I am thankful to **Mr. Punith Kumar, Research Scholar, Department of Computer Science and Engineering, UVCE,** for his constant support during the course of the project.

I express my sincere thanks to all teaching and non-teaching staff, Department of Computer Science and Engineering, UVCE for all the facilities that they have provided me for successfully completing this project.

I also thank my parents and friends for their continuous support and encouragement.

GOURAV KUMAR SHARMA
(U03NM21T029013)

ABSTRACT

"Stock Market Management System" is an Online Stock Trading and Portfolio Management System designed to offer a seamless and user-friendly experience for individuals interested in stock trading and investment portfolio management. In an era marked by the increasing significance of digital finance, the project harnesses the power of robust database management to create an intuitive platform for stock trading enthusiasts. This project provides essential features, including user registration and authentication, a secure wallet system for efficient balance management, real-time monitoring of stock prices, and the ability to buy and sell shares across a diverse range of companies. Users can effectively oversee their investment portfolios, explore detailed stock information, and make informed investment decisions through access to up-to-date stock market data. The project's graphical user interface (GUI) is developed using the CustomTkinter library in Python, ensuring accessibility for users with varying levels of technical expertise. The MySQL database serves as a reliable repository for user data, transaction history, and comprehensive stock market information. This project represents a practical embodiment of database management, data retrieval, and user interface design, tailored to the realm of online stock trading. It serves as a valuable tool for those seeking to engage with the intricacies of stock markets, make equity investments, and effectively manage their investment portfolios. With a strong emphasis on user convenience, financial transparency, and real-time market insights, It emerges as a significant advancement in the domain of digital stock trading and investment portfolio management, offering a dynamic and responsive platform for users to navigate the complexities of financial markets.

CONTENTS

CHAPTER 1

INTRODUCTION

This chapter will discuss the various features and aim of Stock Market Management system

1.1 Background and Context

In the context of an increasingly digital and financially aware world, the "Stock Market Management System" project emerges as a valuable application aimed at educating users about stock trading and investment while providing a practical, user-friendly platform for managing virtual portfolios. The stock market is a complex and dynamic domain, and individuals, especially beginners, can benefit from a controlled environment for learning and gaining experience.

1.2 Objective

The objective of the "Stock Market Management System" project is to develop an online platform that facilitates stock trading and portfolio management in a secure, user-friendly environment. The project aims to educate users about stock trading and investment through practical, hands-on experience in a risk-free setting. Key objectives include:

1. **Educational Platform**: To provide a comprehensive learning environment where users can understand the fundamentals of stock trading, investment strategies, and portfolio management.
2. **User-Friendly Interface**: To create an intuitive and accessible interface that simplifies the process of stock trading and portfolio management for users with varying levels of experience.
3. **Real-Time Data**: To integrate real-time stock market data, allowing users to monitor and analyze stock performance accurately.
4. **Secure Transactions**: To implement robust security measures to protect user data and ensure the integrity of financial transactions within the platform.
5. **Simulated Trading Environment**: To offer a virtual trading environment where users can practice buying and selling stocks without financial risk, thereby gaining practical experience.

6. **Portfolio Management Tools**: To develop tools that enable users to manage their virtual portfolios, track performance, and make informed investment decisions.
7. **Accessibility**: To ensure the platform is accessible through desktop interfaces, providing users with convenient access to stock market information and trading activities.

1.3 Functionality

- Users register by providing personal information (full name, Aadhar number, PAN card, phone number, password, initial balance).
- Users log in with their credentials.
- Users view and manage wallet balances.
- Users add funds to their wallets.
- Users view and manage their stock portfolios.
- Users buy and sell stocks using real-time market data.
- Educational resources available for users to learn about stock trading.
- Dark and light mode options for user interface customization.
- Security protocols to protect user data and transactions.
- Integration of real-time stock market data.
- User-friendly design and responsive layouts.
- Robust error handling and data validation.
- Reliable database management system (MySQL) for storing user profiles, transaction history, and stock data.
- Comprehensive documentation and help resources.
- Thorough testing and quality assurance.
- Secure deployment and distribution of the application.

1.4 Database Management System

DBMS is a collection of programs that enables users to create and maintain a database The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating and sharing databases among various users and applications. It also provides protection and the security to the database. In case of multiple users, it also maintain the data consistency.

A Relational database is a database that has a collection of tables of data items, all of which is formally described and organized according to the relational model. Data in a single table represents a relation, from which the name of the database type comes. Relation does not contain the duplicate tuples and the tuples of a relation have no specific order. In typical solutions, tables may have additionally defined relationships with each other. In the relational model, each table schema must identify a column or group of columns, called the primary key, to uniquely identify each row. A relationship can then be established between each row in the table and a row in another table by creating a foreign key, a column or group of columns in one table that points to the primary key of another table.

1.4.1 Characteristics of Database Management Systems

- Self-describing nature.
- Keeps a tight control on data redundancy.
- Enforces user defined rules to ensure that integrity of table data.
- Provides insulation between Programs and data, Data abstraction.
- Supports multiple views of the data.
- Helps sharing of data and Multi-user transaction processing.

1.4.2 Advantages of DBMS

- Controlling the redundancy.
- Restricting unauthorized access.
- Providing persistent storage for program objects.
- Providing storage structures for efficient query processing.
- Providing multiple users interfaces
- Representing complex relationships among data.
- Enforcing integrity constraints.

1.5 MYSQL

MySQL is an Oracle-backed opensource relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an opensource enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server. MySQL has the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

The ANSI standard SQL provides basic functions for data manipulation, transaction control, and record retrieval from the database. However, most end users interact with Oracle through applications that provide an interface that hides the underlying SQL and its complexity.

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

1. CREATE

This command is used to create a table or view by giving it a name and specifying its attributes and constraints. The attributes are specified first, and each attribute is given a name, a data type to specify its domain values, and any attribute constraints such as NOT NULL.
Syntax: CREATE TABLE <TNAME>(ATR1 TYP1 CONST1, ATR2 TYP2 CONST2,…)

2. ALTER

The definition of a base table can be altered by ALTER command which is a Schema Evolution command. The possible ALTER TABLE includes adding or dropping a column (attribute), changing a column definition, and adding or dropping table constraints.
Example: ALTER TABLE STUDENT ADD NAME VARCHAR (12)

3. DROP

If a whole schema is not needed any more, the DROP SCHEMA command can be used. There are two drop behaviour options: CASCADE and RESTRICT.

- CASCADE option is used to remove the database schema and all its tables, domains and other elements.
- If the RESTRICT option is chosen in place of CASCADE, the schema is dropped only it has no elements in it; otherwise, the DROP command will not be executed.

  Syntax: DROP TABLE STUDENT CASCADE

1.5.1 Statements in SQL:

Following are the important statements used in SQL.

1. SELECT - Used to retrieve the information from the relation.
2. INSERT - Used to insert the new values to the relation.
3. DELETE - Used to delete one or more existing tuples from the relation.
4. UPDATE - Used to update already existing values in the relation.

1.5.2 Aggregate Functions in SQL:

Following aggregate functions are provided by the SQL.

1) COUNT - Returns number of tuples.
2) SUM - Returns sum of entries in a column.
3) MAX - Returns Maximum value from an entire column.
4) MIN - Returns Minimum value from an entire column.
5) AVG - Returns Average of all the entries in a column.

1.5.3 Constraints in SQL:

Following constraints are provided by the SQL.

1) NOT NULL - Column should contain some value.
2) PRIMARY KEY - Should not allow duplicate and null values to a column.
3) UNIQUE - Each value of a column should be unique.

CHAPTER 2

LITERATURE REVIEW

This chapter focuses on the existing systems for stock market management and establishes the software requirements for the project.

2.1 Survey of existing system

The advent of online stock trading platforms has reshaped the financial industry, democratizing market access and fostering greater participation among individual investors. Extensive research has probed into the proliferation and influence of these platforms, underlining their pivotal role in facilitating market accessibility and enhancing user engagement. Noteworthy investigations have scrutinized the usability and user experience of online trading interfaces, underscoring the imperatives of user-friendly design, real-time data integration, and robust security measures.

- **Existing Systems and Methods**

  Historically, stock trading operations relied heavily on manual data management practices. Processes often entailed the use of rudimentary tools like Excel spreadsheets or paper-based systems to track user portfolios, transaction records, and market data. While these methods exhibited efficacy within certain parameters, they were susceptible to errors and inefficiencies, particularly when managing voluminous datasets.

- **Disadvantages:**

  1. Increased reliance on human resources
  2. Repetitive procedural workflows
  3. Vulnerabilities in security protocols
  4. Instances of data redundancy
  5. Challenges associated with data handling and processing
  6. Complexity in data updating procedures
  7. Tediousness in record-keeping tasks

- **Developed System**

The "Stock Market Management System" project opts for a Python-based solution, leveraging the Tkinter library for both frontend and backend development. Python, renowned for its versatility and efficiency, serves as an ideal programming language for building robust applications. Tkinter, a standard GUI toolkit for Python, facilitates the creation of intuitive and user-friendly interfaces. The system capitalizes on active internet connectivity to seamlessly integrate real-time data feeds and trading functionalities. By streamlining user profile management, transaction recording, and market data analysis, the developed system aims to supplant manual data management practices, enhancing operational efficiency and data accuracy.

2.2 SOFTWARE REQUIREMENT

Operating System: Windows/Linux/MacOS

Frontend: Tkinter (Python)

Backend: Python

Database: MySQL

- **Frontend:**

Tkinter: Tkinter stands as the de facto GUI library for Python, offering a rich set of tools for building responsive and visually appealing interfaces. Its versatility and simplicity make it an optimal choice for developing desktop applications.

- **Ease of Use:** Tkinter's intuitive syntax and design facilitate rapid application development.
- **Widgets:** The library provides a diverse array of widgets for constructing dynamic user interfaces, including buttons, labels, text fields, and more.
- **Geometry Management:** Tkinter's geometry managers enable precise control over widget placement and layout, ensuring aesthetically pleasing UI designs.
- **Event Handling:** Tkinter supports event-driven programming, allowing seamless integration of user interactions with application logic.

- **Cross-Platform Compatibility:** Applications developed with Tkinter are compatible across major operating systems, ensuring broad accessibility.
- **Canvas Widget:** Tkinter's Canvas widget offers versatile graphics capabilities, ideal for visualizing real-time stock data and market trends.

➢ Backend:

1. **Python:** Python's versatility and extensive ecosystem of libraries make it well-suited for backend development tasks. Its simplicity and readability expedite development cycles while ensuring robust performance.

    - **Advantages of Python:**
    - Versatility: Python's broad applicability extends to both frontend and backend development.
    - Simplicity: Python's straightforward syntax enhances code readability and maintainability.
    - Community Support: The vast Python community provides access to an abundance of resources and frameworks.
    - Integration: Python seamlessly integrates with diverse databases and third-party APIs, facilitating efficient data management and processing.

2. **MySQL:** MySQL emerges as the preferred database management system for the "Stock Market Management System" project, owing to its reliability, scalability, and robust feature set. MySQL's ACID compliance, data security measures, and high-performance capabilities position it as an ideal solution for managing user profiles, transaction records, and real-time market data.

By harnessing the capabilities of Python with Tkinter for frontend development and MySQL for backend data management, the "Stock Market Management System" project aspires to deliver a comprehensive and user-centric desktop application for stock trading and portfolio management. This integrated approach ensures optimal performance, data security, and an intuitive user experience, aligning with the project's overarching objective

CHAPTER 3

PROPOSED WORK

This chapter will discuss the proposed work for the Stock Market Management system

3.1 Entity Relationship (ER) model:

An entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities.

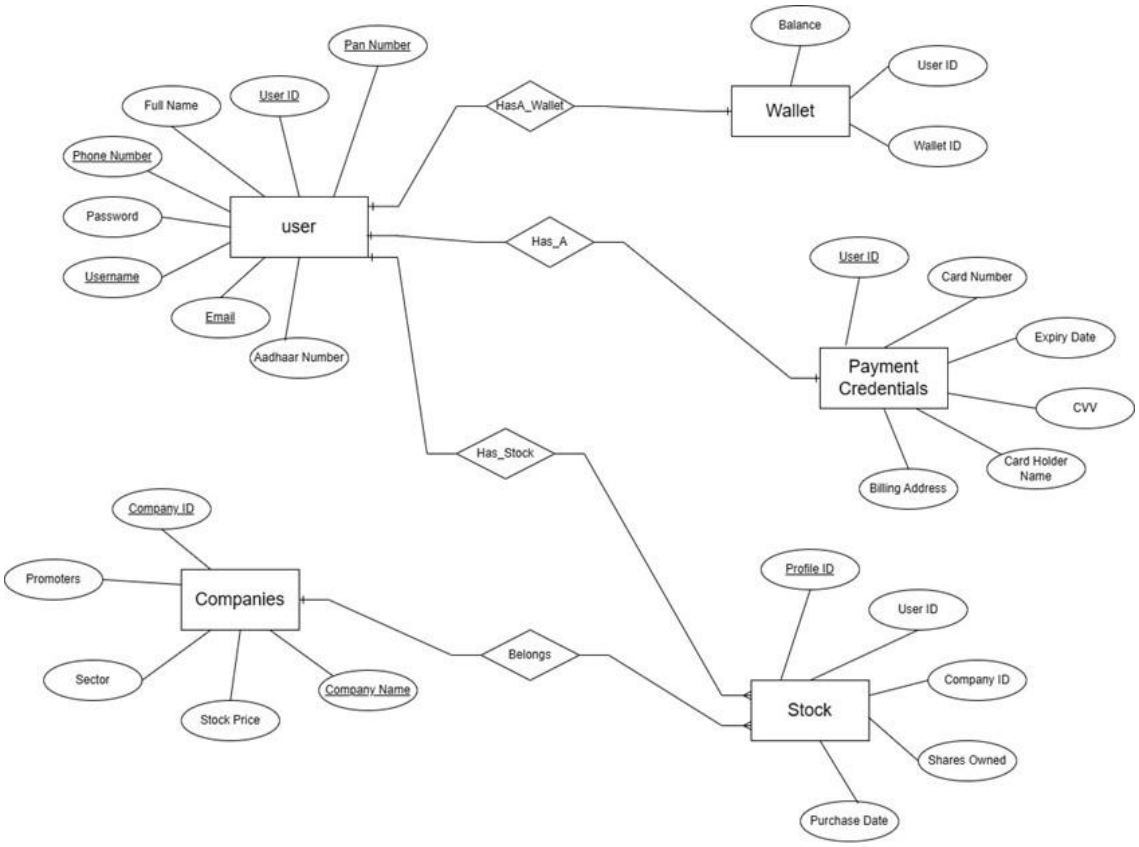

Figure 3.1 : ER Model

Entities and their attributes:

In this project, we will be using several entities to manage user information, wallets, payment credentials, companies, and stocks. Each of these entities plays a crucial role in the financial record management process.

User is the primary entity in the financial record, representing the users of the system. The fields included are User ID, Full Name, Phone Number, Password, Username, Email, Aadhaar Number, and Pan Number where User ID is the primary key of this table.

Wallet is another important entity, representing the user's wallet details. The system will allow for the input and tracking of wallet information, such as Wallet ID, Balance, and User ID where Wallet ID is the primary key.

Payment Credentials is an entity that keeps track of the user's payment information. The fields included are Card Number, Expiry Date, CVV, Billing Address, Card Holder Name, and User ID where Card Number is the primary key.

Companies is another important entity in the financial record, representing the details of companies. The fields included are Company ID, Promoters, Sector, Stock Price, and Company Name where Company ID is the primary key.

Stock is an entity that represents the details of stock owned by users. The fields included are Profile ID, User ID, Company ID, Shares Owned, and Purchase Date where Profile ID is the primary key.

All these entities will be stored in the MySQL database and will be connected to each other through relationships. This allows for easy retrieval and analysis of information, as well as the ability to generate financial reports and statistics. By effectively managing these entities, the Financial Record Management System will streamline the financial management process and provide valuable insights and data for users.

3.2 Relational model:

The following is the relational model for the proposed management system:

USER

User_ID	Full_Name	Phone_Number	Password	Username	Email	Aadhaar_Number	Pan_Number

WALLET

Wallet_ID	Balance	User_ID

PAYMENT_CREDENTIALS

Card_Number	Expiry_Date	CVV	Billing_Address	Card_Holder_Name	User_ID

COMPANIES

Company_ID	Promoters	Sector	Stock_Price	Company_Name

SHARES

Share_ID	User_ID	Company_ID	Shares_Owned	Purchase_Date

3. 3 Normalization

1st Normal Form (1NF)

All the entities in the system such as User, Wallet, Payment Credentials, Companies, and Stock are in 1NF. This is because each record has a unique primary key and each field contains atomic (indivisible) values.

2nd Normal Form (2NF)

All the entities in the system are in 2NF. This is because all non-primary key fields are functionally dependent on the primary key. This means that each field in the table is dependent on the primary key and not on any other non-primary key field.

3rd Normal Form (3NF)

All the entities in the system are in 3NF. This is because there are no transitive dependencies between non-primary key fields. This means that all non-primary key fields are directly dependent on the primary key, and there are no dependencies between non-primary key fields.

For example, the **User** entity has a primary key of **User_ID**, and the Full_Name, Phone_Number, Password, Username, Email, Aadhaar_Number, and Pan_Number fields are dependent on the primary key. Similarly, the **Wallet** entity has a primary key of **Wallet_ID**, and the Balance and User_ID fields are dependent on the primary key. The **Payment Credentials** entity has a primary key of **Card_Number**, and the Expiry_Date, CVV, Billing_Address, Card_Holder_Name, and User_ID fields are dependent on the primary key. The **Companies** entity has a primary key of **Company_ID**, and the Promoters, Sector, Stock_Price, and Company_Name fields are dependent on the primary key. Finally, the **Stock** entity has a primary key of **Profile_ID**, and the User_ID, Company_ID, Shares_Owned, and Purchase_Date fields are dependent on the primary key.

Overall, the database for this project is in 3rd Normal Form, which ensures that the data is organized in a manner that minimizes data redundancy and improves data integ

CHAPTER 4

RESULTS

The Stock Management System was implemented and tested using sample data. The results of the implementation show that the system effectively keeps track of user information, wallet balances, payment credentials, company details, and stock ownership, providing an easy-to-use interface for users.

The system's user interface was evaluated by a group of test users, and they found it to be user-friendly and easy to navigate. A well-designed Stock Management System is capable of streamlining the stock management process and providing valuable insights and data for users. In recent years, there has been a growing interest in the development of computerized systems for stock management to enable fast and efficient sharing of critical information across all relevant entities. The system's database was found to be well-designed and normalized, providing good data integrity and consistency.

Overall, the Stock Management System was found to be a powerful tool that streamlines the stock management process and provides valuable insights and data for users. The system is expected to significantly improve the efficiency and accuracy of stock management and provide a better experience for all parties involved.

4.1 Screenshots

The following is a series of screenshots of the developed application.

- **Figure 4.1:** Login page
- **Figure 4.2:** Sign up page
- **Figure 4.3:** User’s Page
- **Figure 4.4:** Buy/Sell
- **Figure 4.5:** Quantity

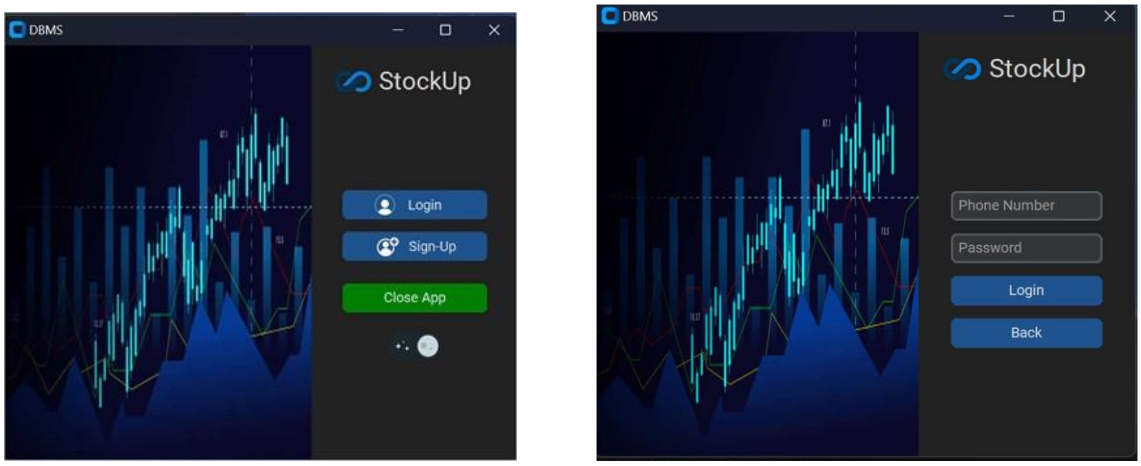

Figure 4.1: Login page

Figure 4.2:Sign up page

Figure 4.3: User's Page

Figure 4.4: Buy/Sell

Figure 4.5: Quantity

Figure 4.1: Login Page

Figure 4.1 is the login page screen that appears to every user where the user has to login with their respective User ID and password.

Figure 4.2: Sign-up Page

Figure 4.2 is the sign-up page that allows new users to create an account by providing necessary information such as full name, email, phone number, and password.

Figure 4.3: User's Page

Figure 4.3 is the user's page that shows the user's wallet balance, payment credentials, and stocks owned. This page provides an overview of the user's financial status and stock portfolio.

Figure 4.4: Buy/Sell

Figure 4.4 is the buy/sell page where users can execute transactions to buy or sell stocks. This page includes options to select the stock, enter the quantity, and confirm the transaction.

Figure 4.5 Quantity

Figure 4.5 is the quantity page that shows the quantity of each stock owned by the user. This page provides detailed information about the shares held by the user for each company in their portfolio.

CONCLUSION

The "Stock Market Management System" project offers a comprehensive and user-friendly platform for simulated stock trading and portfolio management. Key features include user registration, authentication, wallet management, and portfolio tracking, all designed to provide a realistic trading experience.

The project has successfully laid the groundwork for future enhancements such as real-time data integration, mobile applications, advanced trading features, and educational resources. These improvements can further enrich the user experience and expand the application's capabilities.

In conclusion, "Stock Market Management System" is a valuable educational tool for individuals interested in stock trading and investment. It provides practical experience in a risk-free environment and has the potential for continuous growth and improvement, aligning with user needs and the evolving financial market.

The "Stock Market Management System" project has laid a robust foundation for stock trading and portfolio management, with many opportunities for future enhancements. User profiles can be expanded to include personal information, investment preferences, and risk tolerance, and users could link real brokerage accounts for live trading. Integrating real-time market data providers will enable features like live stock quotes, charts, and news feeds.

Advanced trading options, such as limit and stop-loss orders, along with technical analysis tools, will provide users with more sophisticated trading capabilities. Machine learning can be integrated to offer stock recommendations and predictions, enhancing the decision-making process for users.

Developing mobile applications for iOS and Android will increase accessibility, ensuring data synchronization between web and mobile platforms. Adding social networking features will enable users to interact, share strategies, and participate in forums and chat rooms

Support for international markets, including multi-currency trading, will broaden the platform's reach. Security can be further enhanced with two-factor authentication and data encryption. Providing educational resources, such as articles, tutorials, and webinars, will help users improve their trading knowledge and skills.

API integration will allow third-party developers to create extensions and integrations, fostering innovation. Gamification elements and reward systems can make the app more engaging. Ensuring compliance with financial regulations is essential, especially if live trading is introduced.

BIBLIOGRAPHY

[1] Constantine A. Balanis, *Antenna Theory: Analysis and Design*, 2nd Edition; John Wiley Sons Inc. New York, 2001,pp12-130

[2] Inder J Gupta, Aharon A Ksienski, "Effect of Mutual Coupling on the Performance of Adaptive Arrays" *IEEE Trans. Antennas Propagat*., vol. AP-31, no.5, pp.785-761, Sept. 1663

[3] S. Chen, L. Hanzo, N.N. Ahmad, and A. Wolfgang "Adaptive minimum bit error rate beam forming assisted QPSK receiver," *Proc. IEEE International Conference on Communications*, June 2004, Paris, France

[4] J. Lundback, S. Nordebo, "Analysis of a tripole array for polarization and direction of arrival estimation" *Proc. IEEE Sensor Array and Multichannel Signal Processing Workshop Proceedings*, 2004, 18-21, July 2004, pp. 284 - 288

[5] G.Brandli and M. Dick, "Alternating current fed power supply," U.S.Patent 4 084 217, Nov. 4, 1678

[6] E. E. Reber, R. L. Mitchell, and C. J. Carter, "Oxygen absorption in the Earth's atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (4230-46)-3, Nov. 1668

[7] M. Duncan. "Engineering Concepts on Ice. Internet: www.iceengg.edu/staff.html, Oct. 25, 2000 [Nov. 26, 2003].